

Lab #4b:
CLIENT-SIDE SCRIPTING (JSON and jQuery)

Topic	Web page development using JSON and jQuery
Domain of Learning	Psychomotor (P2: Set; P3: Guided Respond; P4: Mechanism)
Learning objective	Students will be able to retrieve, parse, and display JSON data using jQuery, applying fundamental client-side scripting techniques to dynamically update web content.
Lab activity objective	Students will demonstrate their ability to load JSON data, extract specific data points, and use jQuery to dynamically update HTML elements based on the data, creating an interactive and user-friendly web experience.

Instruction: Answer all questions. Write your answer and screenshot the output in Microsoft Word. Push all codes to your GitHub repository and include your GitHub repository link.

Submit through Author in PDF format.

1. Consider the following collection of books:

	Book 1	Book 2
title	The Great Gatsby	To Kill a Mockingbird
author	F. Scott Fitzgerald	Harper Lee
publicationYear	1925	1960
genres	Fiction, Classics	Fiction, Coming-of-Age

Create a JSON representation (.json file) of this collection. You may refer to the following GitHub repository

https://github.com/ariffinmzin/webdev_week_9_2023/blob/main/students.json as a reference.

Answer:

```
{
  "Books": [
    {
      "title": "The Great Gatsby",
      "author": "F. Scott Fitzgerald",
      "publicationYear": 1925,
      "genres": "Fiction, Classics"
    }
  ]
}
```

```
},
{
    "title":"To Kill a Mockingbird",
    "author":"Harper Lee",
    "publicationYear":1960,
    "genres":"Fiction, Coming-of-Age"
}

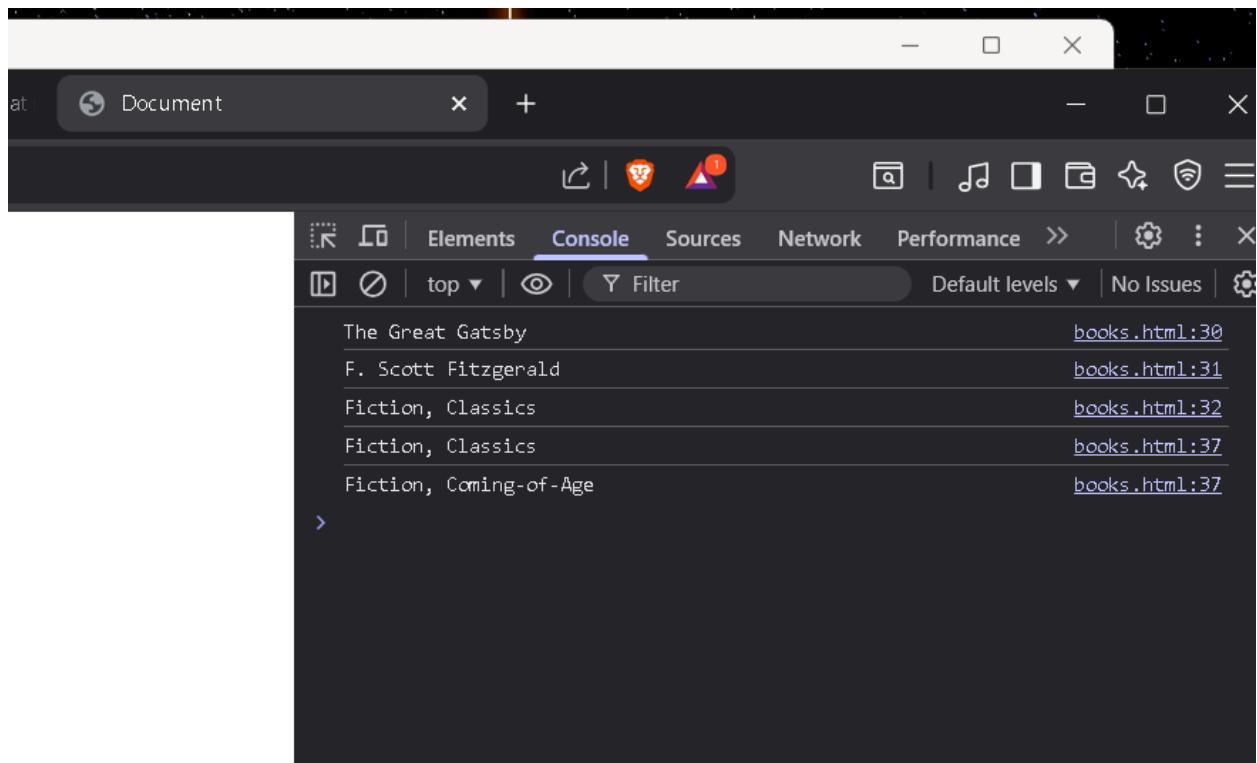
]
```

2. Based on the JSON you have written in Question (1), write an HTML file to display the following details.

- (a) The title of the first book.
- (b) The author of the second book.
- (c) The genres of all books.

You can display the details through `Inspect - Console` on your browser. You may refer to the following GitHub repository
https://github.com/ariffinmzin/webdev_week_9_2023/blob/main/students.html as a reference.

Answer:



3. Rewrite the HTML code below with the jQuery statements that are equivalent to the JavaScript DOM statements in a, b, and c.

```
<!DOCTYPE html>
<html>
<head>
    <title>jQuery DOM Operations</title>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
    <style>
        .myStyle {
            background-color: coral;
            padding: 16px;
        }
    </style>
</head>
<body>
    <div id="myElement">Original Text</div>

    <div id="myDIV">
        <p>I am a myDIV.</p>
    </div>

    <button id="myButton">Click me</button>
<script>
    // a
```

```
// var element = document.getElementById("myElement");
// element.textContent = "Modified Text";

//jQuery equivalent of a - write your answer here

// b
// var paragraph = document.querySelector("#myDIV");
// paragraph.classList.add("myStyle");

//jQuery equivalent of b - write your answer here

// c
// var button = document.getElementById("myButton");
// button.addEventListener("click", function() {
//   alert("Button clicked!");
// });

//jQuery equivalent of c - write your answer here

</script>
</body>
</html>
```

Answer:

```
<!DOCTYPE html>

<html>
<head>
<title>jQuery DOM Operations</title>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
<style>
.myStyle
{
background-color: coral;
```

```
padding: 16px;  
}  
  
</style>  
  
</head>  
  
<body>  
  
<div id="myElement">Original Text</div>  
  
<div id="myDIV">  
  
<p>I am a myDIV.</p>  
  
</div>  
  
<button id="myButton">Click me</button>  
  
<script>  
  
// This wrapper ensures the code only runs after the page is fully  
loaded.  
  
$(document).ready(function() {  
  
    // a  
  
    // var element = document.getElementById("myElement");  
  
    // element.textContent = "Modified Text";  
  
    //jQuery equivalent of a  
  
    $("#myElement").text("Modified Text");  
  
    // b  
  
    // var paragraph = document.querySelector("#myDIV");  
  
    // paragraph.classList.add("myStyle");
```

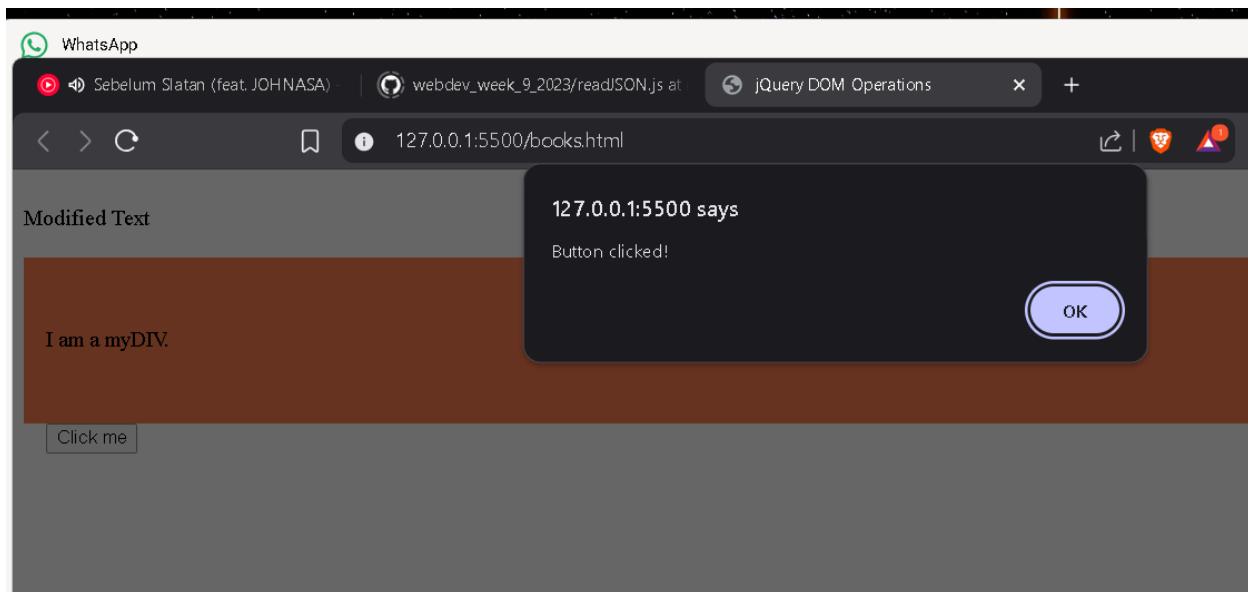
```
//jQuery equivalent of b
$( "#myDIV" ).addClass("myStyle");

// c
// var button = document.getElementById("myButton");
// button.addEventListener("click", function() {
//   alert("Button clicked!");
// });

//jQuery equivalent of c
$("#myButton").click(function() {
  alert("Button clicked!");
});

}); // <-- Closes the $(document).ready()

</script>
</body>
</html>
```



Assessment:
A. JSON Data Handling and jQuery Integration (CLO2 - Psychomotor)

Criteria	Excellent (5)	Good (4)	Satisfactory (3)	Needs Improvement (1-2)	Unsatisfactory (0)	Marks
Parsing JSON Data (P4)	Parses JSON data accurately and efficiently, demonstrating a thorough understanding of data structures.	Parses JSON data with minor issues, demonstrating basic understanding.	Basic JSON parsing is present but lacks depth or efficiency.	Limited ability to parse JSON data accurately, requires guidance.	No JSON parsing applied in the code.	___/5 x 5
Manipulating JSON Data (P4)	Skillfully manipulates JSON data, displaying specific data points or transforming data as needed.	Manipulates JSON data effectively with minor improvements needed.	Basic data manipulation is present but lacks complexity or relevance.	Minimal data manipulation, limited comprehension of JSON structure.	No JSON data manipulation present.	___/5 x 5
jQuery Integration for Displaying Data (P4)	jQuery is used effectively to dynamically display JSON data on the web	jQuery is used to display JSON data with minor issues in dynamic	Basic jQuery is used to display JSON, lacks interactivity.	Minimal use of jQuery for displaying JSON,	No jQuery integration for JSON display.	___/5 x 5

	page, creating a user-friendly interface.	content presentation.		limited interaction.		
Total						/15

B. Presentation and Explanation of JSON and jQuery Concepts (CLO3 - Affective)

Criteria	Excellent (5)	Good (4)	Satisfactory (3)	Needs Improvement (1-2)	Unsatisfactory (0)	Marks
Clarity and Confidence (A1)	Presents the JSON and jQuery concepts clearly and confidently, engaging the audience effectively.	Generally clear with minor confidence issues, demonstrates basic understanding.	Presentation is adequate but lacks confidence or depth.	Struggles to explain JSON and jQuery concepts confidently.	Presentation is difficult to follow, lacks clarity.	___/5 x 2
Explanation of JSON and jQuery Integration (A2)	Demonstrates a deep understanding of JSON structure and jQuery integration, explaining choices effectively.	Good understanding of JSON and jQuery integration with clear justification.	Basic explanation of JSON and jQuery, minimal depth.	Limited understanding, struggles to explain purpose or integration choices.	Fails to explain JSON and jQuery integration effectively.	___/5 x 3
Total						/5